(Programming in C# and Entity Framework)

Day 1:

Module 1: C# Language Concepts

• Lesson 1: Methods

• Lesson 2: Method Overloading

• Lesson 3: Exception Handling

Lab

Module 2: C# Structures, Collections and Events

• Lesson 1: Structs

• Lesson 2: Enums

• Lesson 3: Built-in Collections

• Lesson 4: Events

Labs

Module 3: C# Classes

• Lesson 1: Creating Classes

• Lesson 2: Interfaces

• Lesson 3: Understanding Generics in C#

Labs

Day 2:

Module 4: C# Inheritance

• Lesson 1: Hierarchies of Classes

• Lesson 2: Polymorphism

• Lesson 3: Extending Classes

Labs

Module 5: Input and Output

• Lesson 1: File I/O

• Lesson 2: Serialization and Deserialization

• Lesson 3: Streams

Labs

Day 3:

Module 6: Database Access

• Lesson 1: Entity Framework

• Lesson 2: Relationships in Entity Framework(1:1,1:M,M:M)

• Lesson 3: Load Data(Eager Loading, Explicit Loading, Lazy Loading)

• Lesson 4: Fluent Api

• Lesson 5: Constraints

Day 4:

• Lesson 6: Keyless Entities

• Lesson 7: Save Data

• Lesson 8: Change Tracking

• Lesson 9: Migration

• Lesson 10: Scaffolding

• Lesson 11: Logging

Labs

Day 5:

Module 8: Using the Network

• Lesson 1: Web Services

• Lesson 2: REST and OData

• Lesson 3: ASP.NET Core MVC

Labs

Module 9: Graphical User Interfaces

• Lesson 1: Using UI Frameworks

• Lesson 2: Data binding

• Lesson 3: Styling the UI

Labs

Day 6:

Module 9: Application Performance

• Lesson 1: Understanding tasks

• Lesson 2: Linking tasks

• Lesson 3: Async and await

• Lesson 4: Concurrent tasks

Labs

Module 10: Designing for Reuse

• Lesson 1: Reflection

• Lesson 2: Attributes

(ASP.net Core and ASP.net Core Web API)

Day 7:

Module 01: Exploring ASP.NET Core

- Introducing Microsoft Web Technologies
- Getting Started with Razor Pages in ASP.NET Core
- Introducing ASP.NET Core MVC

Labs

Module 02: Using Razor Pages and Middleware

- Configuring Middleware
- Configuring Services

Labs

Module 03: Developing Controllers

- Writing Controllers and Actions
- Configuring Routes
- Writing Action Filters

Day 8:

Labs

Module 04: Developing Views

- Creating Views with Razor Syntax
- Using HTML Helpers and Tag Helpers
- Reusing Code in Views

Labs

Module 05: Developing Models

Developing Models
Passing Models to Views
Binding Views to Model Classes and Displaying Data
What Are Model Binders?
Adding CRUD Operations to Controllers

Labs

Day 9:

Module 06: Using Entity Framework Core in ASP.NET Core

- Working with Entity Framework Core
- Using Entity Framework Core Database Providers

Labs

Module 07: Using Layouts, CSS and JavaScript in ASP.NET Core

- Using Layouts
- Using CSS
- Using JavaScript

Labs

Module 08: Client-Side Development

- Responsive Web Design
- Using Front-end Development Tools

Labs

Day 10:

Module 09: Testing and Troubleshooting

- Testing MVC Applications
- Implementing an Exception Handling Strategy
- Logging MVC Applications

Labs

Module 10: Managing Security

- Authentication in ASP.NET Core
- Authorization in ASP.NET Core
- Defending from Common Attacks

Labs

Day 11:

Module 11: Performance and Communication

- Implementing a Caching Strategy
- Managing State
- Supporting Two-Way Communication

Labs

Module 12: Implementing Web APIs

- Introducing Web APIs
- Developing a Web API
- Calling a Web API

Labs

(Microservices)

Day 12:

Module 13: Microservice

- Understanding monolithic application and microservice
- Designing Microservice
- First Microservice

Labs

Module 14: Microservice Communication

- Synchronous inter-service communication
- Rabbit MQ Integration
- Asynchronous inter-service communication

Labs

Day 13:

Module 15: Main Application to call microservices

- Initial Frontend Integration
- ASP.net core MVC as frontend application

Labs

Module 16: Security

- Identity in Microservices
- Microservices Security Patterns and Techniques
- Implementing Microservices Security

Labs

Day 14:

4. Service Discovery & API Gateway (Intro)
    - Why API Gateway is needed
    - Quick setup with **Ocelot**

Labs

5. Containerization
    - Basics of Docker
    - Running microservices in Docker containers